



# Annotation Graphs: A Graph-Based Visualization for Meta-Analysis of Data based on User-Authored Annotations

Jian Zhao, Michael Glueck, Simon Breslav, Fanny Chevalier, Azam Khan

## ► To cite this version:

Jian Zhao, Michael Glueck, Simon Breslav, Fanny Chevalier, Azam Khan. Annotation Graphs: A Graph-Based Visualization for Meta-Analysis of Data based on User-Authored Annotations. IEEE Transactions on Visualization and Computer Graphics, 2016, 10.1109/TVCG.2016.2598543 . hal-01353368

**HAL Id: hal-01353368**

**<https://inria.hal.science/hal-01353368>**

Submitted on 14 Dec 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Annotation Graphs: A Graph-Based Visualization for Meta-Analysis of Data based on User-Authored Annotations

Jian Zhao, Michael Glueck, Simon Breslav, Fanny Chevalier, and Azam Khan



Fig. 1. User-authored annotations are visualized as an annotation graph, including three different layouts a) the projection-plot, b) the slice-plot, and c) the circular-plot, each enabling an analyst to browse the annotation semantics from a different perspective. In the annotation graph, nodes reflect free-form comments (red), short word tags (blue), and their associated data selections (green); links represent relations between annotations. Together, the content and relations of annotations encode annotation semantics.

**Abstract**—User-authored annotations of data can support analysts in the activity of hypothesis generation and sensemaking, where it is not only critical to document key observations, but also to communicate insights between analysts. We present *annotation graphs*, a dynamic graph visualization that enables meta-analysis of data based on user-authored annotations. The annotation graph topology encodes annotation semantics, which describe the content of and relations between data selections, comments, and tags. We present a mixed-initiative approach to graph layout that integrates an analyst’s manual manipulations with an automatic method based on similarity inferred from the annotation semantics. Various visual graph layout styles reveal different perspectives on the annotation semantics. Annotation graphs are implemented within C8, a system that supports authoring annotations during exploratory analysis of a dataset. We apply principles of Exploratory Sequential Data Analysis (ESDA) in designing C8, and further link these to an existing task typology in the visualization literature. We develop and evaluate the system through an iterative user-centered design process with three experts, situated in the domain of analyzing HCI experiment data. The results suggest that annotation graphs are effective as a method of visually extending user-authored annotations to data meta-analysis for discovery and organization of ideas.

**Index Terms**—Externalization, user-authored annotation, exploratory sequential data analysis, graph-based visualization.

## 1 INTRODUCTION

Creating annotations is an essential activity when making sense of data during exploratory analysis. It has been identified as one of the “critical tasks that enable iterative visual analysis” [17]. In Exploratory Sequential Data Analysis (ESDA), which focuses on sequential data such as time-series and event streams, “annotate” is highlighted as a key step performed by analysts [29]. User-authored annotations can indeed support the process of generating hypotheses, verifying conjectures, and deriving insights, where it is not only critical for analysts to document key observations, but also to communicate findings with others.

Facilitating annotations is an important aspect of supporting the knowledge externalization process [10, 20, 21, 24]. Yet, commercial exploratory data analysis systems and visualizations that support annotations still present user-authored comments as lists; in a linear and textual form. This approach may be too rigid for data sensemaking—a process which is typically nonlinear and requires analysts to navigate through details of a dataset, comparing and revisiting different data segments and constantly re-evaluating questions in the quest for patterns and knowledge discovery [28]. To cope with such complexity in the exploration process, recent research advocates for a rich range of

annotations, including comments in both free-text and tag formats, relating comments and data selections, and accessing all annotations in a holistic view [6, 36]. For example, in prior work we applied an open-source ESDA tool [5] to analyze simulations of building occupants and found that rich annotation support helped practitioners better understand simulation models and communicate findings [6].

In this paper, we build on these recommendations and propose a new direction. In particular, we focus on supporting *meta-analysis* of data by leveraging the annotations users create during exploratory data analysis. Our approach visualizes annotations as a dynamic graph that serves as a visual summary of all user-authored annotations of the data. Nodes represent annotations, including free-form comments, short word tags, and data selections. Links encode the relations between annotations. Together, the content of annotations and the relations between them encode *annotation semantics* (e.g., explicit co-occurrence, implicit similarity). By interactively manipulating the graph layout and inspecting its topology, meta-analysis of data can be performed. For example, the graph may form clusters of similar concepts, expose related observations, or isolate particular tags or selections. We intuit that visual exploration of user-authored annotations could help organize thoughts and promote discovery.

While user-authored annotations have been represented using graphs in the literature, these systems do not fully support interactive visual exploration of the graph topology. Graph layouts are either based on purely manual manipulations (e.g., [24]), which become cumbersome as the number of annotations grows, or rely on purely automatic computations (e.g., [9]), which lack flexibility. To effectively support meta-analysis, we employ a mixed-initiative approach that enables

- Jian Zhao, Michael Glueck, Simon Breslav, and Azam Khan are with Autodesk Research. Emails: [firstname.lastname@autodesk.com](mailto:firstname.lastname@autodesk.com).
- Fanny Chevalier is with INRIA. Email: [fanny.chevalier@inria.fr](mailto:fanny.chevalier@inria.fr).

analysts to manipulate the annotation graph. That is, specific nodes can be positioned manually, while other nodes are placed automatically according to a customizable similarity metric that takes into account both annotation semantics, as well as manually-positioned nodes.

To implement and assess annotation graphs, we extended an open-source interactive data annotation application [5] into a system we call C8. We ground the design in the principles of ESDA [29] and contribute a mapping of ESDA concepts to Brehmer & Munzner's multi-level task typology [4]. We situated a design study in the domain of analyzing temporal or sequential data observations captured in HCI experiments. The design of annotation graphs was refined through an iterative user-centered design process with three experts. To evaluate annotation graphs, including our novel mixed-initiative layout approach, we engaged these three experts in a deployment study.

Our results with these HCI usage scenarios demonstrate that annotation graphs, and the mixed-initiative layout, are promising in helping analysts document findings, organize thoughts, and consolidate ideas. Since temporal data is ubiquitous in many real-world applications, we believe that meta-analysis using annotation graphs and the C8 workflow of selecting data, commenting, and tagging, are generalizable and could be applied to exploratory data analysis in other domains.

## 2 RELATED WORK

Kang and Stasko recently demonstrated the importance of externalizing an analyst's thought process, of which making annotations is a key aspect [21]. They observed that analysts continuously build a conceptual model in the sensemaking process and suggested visual representations can be leveraged to help analysts browse the history of previous discoveries in order to "*connect, group, and organize concepts*." Mahyar et al. further suggests externalization is critical in collaborative analysis scenarios, highlighting note-taking using free-form text, charts, and tables as one of the main activities [23]. Studies in intelligence analysis identified schematizing information during data exploration, i.e., organizing findings in a structured format, as a key component for modern visual analytics systems [10, 20].

In the literature, many techniques to facilitate externalization during analysis have been proposed (see Xu et al. [38] for a general review), including analytical techniques to *automatically* tag interaction logs, or interfaces and visualizations to facilitate *manually* authoring annotations. In this work, we focus on manual user-authored annotations in visualizations, building on prior work in interaction trace analysis, authoring annotations, and data meta-analysis from annotations.

### 2.1 Generating Annotations from Interaction Traces

A user's interactions while performing visual analysis can be analyzed to understand the analyst's thought process [13], such as identifying key steps in the data exploration and sensemaking process. Interaction histories can include both low-level inputs (e.g., mouse clicks [31] or eye-tracking data [12]) and high-level actions (e.g., zooming and panning [15]). Researchers have employed interaction logs to automatically capture a user's findings in data exploration and visual analytics (e.g., [15, 16, 30, 39]). While a user's interaction history can shed light on the process of sensemaking, it remains difficult to infer and summarize high-level insights from a series of interactions without more explicit input from a domain expert to impart semantic meaning.

Data exploration is essential to engage domain experts in data investigation and many visualization systems facilitate the exploration of interaction logs, for example, VisTrails [8] and the P-Set model [19] visualize user interactions with a system as a tree or a graph. Others have focused on HCI experiment data, for example, Mimic supports exploring user micro-interactions (e.g., mouse clicks, scrolls) that are obtained in web-based crowd-sourced experiments [7] and VA<sup>2</sup> integrates diverse sources of data (e.g., think-aloud protocols, eye-tracking, interaction logs) to facilitate experimenters with the evaluation of visualizations [2]. We also situate our usage scenario in the domain of HCI experiment data. While these systems support exploration of interaction data, they do not support the creation of annotations. In our work, we present an interactive and visual approach to annotation creation in tandem with exploration.

### 2.2 Support for User-Authored Annotations

User-authored annotations support externalization of high-level insights by capturing an analyst's actual observations. Several systems enable analysts to create annotations inline with data to document hypotheses and findings (e.g., [34]). Another approach enables users to explicitly create abstract concepts and relations in *mental maps* (e.g., [37]).

Some web-based visualization platforms, such as ManyEyes [34] and sense.us [18], incorporate functionality for linking user comments to specific visualization views. ManyEyes implements a discussion board associated with a visualization to collect comments from multiple users. Sense.us further enables users to create graphical annotations (e.g., arrows and free-form drawings) directly over the visualization. A solution for business intelligence supports context-aware annotations linked across multiple views [14]. While each offers great flexibility in adding a variety of comments and observations, these systems do not reveal the underlying relations between individual annotations (e.g., conceptual similarity), which is part of the annotation semantics.

CommentSpace addresses this problem by supporting a fixed vocabulary of tags (i.e., "question", "hypotheses", "to-do") and links (i.e., "evidence-for", "evidence-against") for analysts to organize their comments [36]. PathFinder, targeted at citizen scientists, relies on a similar mechanism [22]. Likewise, SandBox equips several direct manipulation gestures to create annotations in a sticky-note representation and further add tags and links [37].

Enumerating fixed categories for tags and links builds a common vocabulary that can be interpreted by both analyst and underlying system, but imposes rigidity that may inhibit an analyst from expressing the richer semantics of their mental model in the annotations. Moreover, these systems lack visual representations of the annotation space, i.e., they present annotation semantics in a text-based format, making it difficult for analysts to get an overview of the insights found and perform meta-analysis of data on top of the annotations. In an unpublished open-source annotation tool we developed [5] annotations are visualized in a static graph. Annotation graphs extends the concept of graph-based representation of annotations to a dynamic and interactive visualization of the annotation semantics. It is through interaction that we seek to support meta-analysis, enabling analysts to organize annotations and discover high-level insights.

### 2.3 Meta-Analysis Based on Annotations

Building on annotation support, a number of techniques have been proposed to provide structured organizations of user-authored annotations and data evidence, most of which adopt the graph structure. For example, Tivoli employs a free-form whiteboard environment [26] and Aruvi enables users to manually organize notes and their relations on a canvas [31]. Other systems, such as Tree Trellis and Table Trellis support aggregation and comparison of linked free-text statements in trees and tables [11], but are designed for inspecting existing statements rather than assisting an ongoing exploration.

Visualizations have been applied to make annotation semantics more comprehensible and manageable, thus promoting meta-analysis and insight discovery. Evidence Matrices lists hypotheses as rows and evidence as columns, indicating consistent and inconsistent relations between evidence and hypotheses, and providing an overview of current analytic reasoning [1]. However, it is not possible to define relations between hypotheses or evidence.

CLIP employs a node-link diagram for representing user-authored concepts and relations in collaborative data analysis [24]. However, it relies on purely manual organization of user comments, which may not scale to large numbers of annotations.

ManyInsights also visualizes annotations as a graph, but relies on purely automated methods to extract key concepts from data and generate annotations [9]. Reliance on generated annotations without support for user-authored annotations could hinder the sensemaking process, since the algorithm's output may not align with an analyst's mental model.

Annotation graphs overcome these limitations by employing a mixed-initiative approach to graph layout, bridging manual and automatic layout approaches, leveraging the benefits of each.

### 3 DESIGNING ANNOTATION GRAPHS AND C8

To develop the annotation graphs, we employed an iterative user-centered design process. Similar to prior work [2, 6], we selected analysis of HCI experiment data as a concrete application domain and recruited three HCI researchers as our domain experts. Since annotation graphs cannot be evaluated in isolation, we implemented annotation graphs within the design and development of a complete annotation system, called C8. We start by introducing the datasets and analysis tasks of our experts, then discuss the theoretical foundation of our design, and finally describe the practical design process.

#### 3.1 Datasets and Application Domain

Although each expert focused on a different area of HCI research, they all collect large quantities of experimental data and want to discover insights that elicit and inform research directions.

- The first expert conducted an experiment to investigate hand movements in a rapid-aimed pointing task, in which participants were asked to point to targets on a tabletop starting from a fixed position on the table. His data consisted of 3D hand trajectories during the pointing action, captured using a motion tracking system.
- The second expert was interested in studying people’s finger movements when using a trackpad. He carried out an experiment to record all touch interactions (e.g., event types, finger coordinates) initiated by participants during their everyday usage.
- Our third expert was also interested in touch interactions on a trackpad, but focused on finger chording patterns (i.e., finger combinations used in gestures). The expert developed a customized tool to capture interaction logs consisting of gesture type (e.g., scrolling), fingers used, and event timestamps of the gestures.

In all cases, datasets contained a list of observations for different experimental conditions; an observation often included task attributes and several time-series. Detailed data attributes of the observations are shown in Table 1. We selectively chose these experts to cover of a variety of sequential datasets in HCI experiments. By considering numerical time-series and discrete temporal events, in both 2D and 3D configurations, we encompass a range of application domains.

#### 3.2 Design Rationale

Since the principles of ESDA [29] were proposed for analyzing HCI data, we turned to these to guide our design. We further align them with the multi-level typology of visualization tasks [4] (Figure 2), forming a coherent theoretical framework to ground our design of C8. Fulfillment of these design goals are discussed in Section 4 and 5.

##### 3.2.1 Exploratory Sequential Data Analysis

Exploratory Sequential Data Analysis (ESDA) [29] is proposed as a framework for analyzing the observational data recorded in HCI studies, such as system interaction logs or video recordings, which have a sequential nature rooted in user interactions. Building on the tradition of exploratory data analysis [33], ESDA focuses on the sequentiality of the data, encourages analysts to pose questions about the data, and promotes exploration rather than confirmation to develop statements that describe observed patterns. ESDA highlights eight approaches (i.e., the Eight C’s) that preserve the sequential integrity of the data while encouraging insights, as discussed below.

**Conversions** transform data to reveal new patterns. The system should support conversions by allowing for flexible organization of multi-attributed data observations, which can be further focused with constraints. Moreover, time-series attributes should be supported and summarized in a way to be aligned with normal data attributes.

Expert domain	Data attributes (time-series are in <i>italics</i> )
3D Pointing Trajectories	user ID, trial ID, trajectory ID, target distance, target size, target direction, <i>trajectory x-, y-, and z-coordinates</i>
Touchpad Interactions	user ID, finger number, <i>event timestamps, event types, finger x- and y-positions, finger x- and y-velocities</i>
Finger Chording Gestures	user ID, gesture type, finger chord (e.g., 00000-11000: right thumb and index finger), hand, <i>event timestamps</i>

Table 1. Data attributes of observations in the design study.

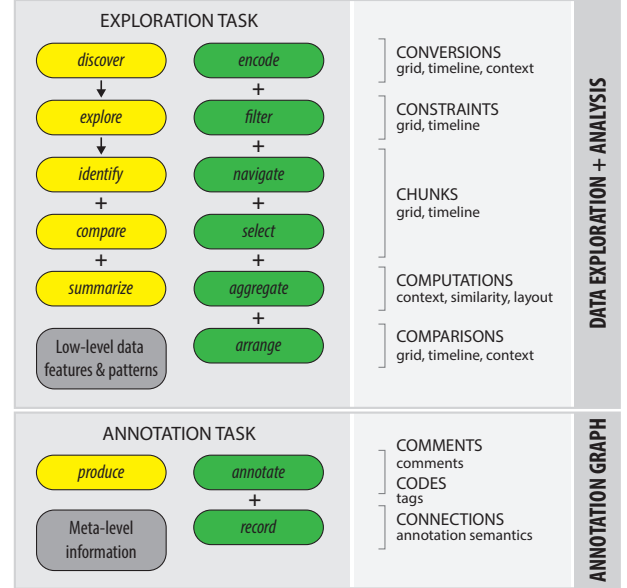


Fig. 2. Schematic representation of the design rationale of C8, grounded with the principles of Exploratory Sequential Data Analysis [29] and Brehmer & Munzner’s multi-level typology of visualization tasks [4] (using their visual language: yellow—why, green—how, and gray—what).

**Constraints** are proposed as a means to focus analysis. Reorganization functions such as sorting, filtering, and grouping of data observations should be supported.

**Chunks** are user-defined sets of data observations that are meaningfully related to the analysis. For example, a certain sequence of user interactions may occur frequently in the observations, such as “search, select, delete”. To allow an analyst to capture these important sequences in the data, chunking should be supported through selection of data observations. These selections should be flexible, for example, encompassing several observations, or merely an interval of a single observation (if time-series data).

**Computations** reduce the data to a summary representation. Selected observations should be aggregated in a manner congruent to the analysis and its application domain. For example, if data represents a 2D position, a large number of selected observations could be aggregated in a heatmap. In addition to the data level, computations should also support meta-analysis, such as calculating similarity of user-authored annotations using the annotation semantics.

**Comparisons** demonstrate the similarities and differences between observations. Various visualization methods should be employed to allow for comparisons at different data scale levels for different purposes (i.e., quick comparison or detailed comparison). The displayed comparative observations should be moderated by the data selection (i.e., chunks) of an analyst. Thus, through the support of constraints, groups of chunks with similar attributes can be explored and compared.

**Comments** provide a means of adding unstructured notes to chunks, and should be the primary method of annotating data in the system. An analyst can use comments to describe the patterns they see in chunks, or pose questions that require further investigation.

**Codes** are labels that can be applied to comments to denote more abstract concepts and meanings. Indicated by previous practices (e.g., [36]), a tagging metaphor should be supported for an analyst to add labels to a single or multiple comments.

**Connections** express the relation among chunks. Our idea of the annotation graph is an embodiment of this concept, presenting chunks (i.e., selections), comments, and codes (i.e., tags) in the context of a graph. This interactive representation supports the exploration of chunks, comments, and codes in the abstract space of an analyst’s annotation semantics, where higher-level meta-analysis can take place.

Hence, our main focus in this study lies in the *connections*, supported by enabling all the previous C’s, which facilitate the meta-analysis phase of data exploration based on annotations.

### 3.2.2 Visualization Tasks

To illustrate the high-level goals and methods involved when users create annotation graphs, we use Brehmer & Munzner’s multi-level typology of visualization tasks [4]. At a coarse level, both *discover* and *produce* goals should be supported (Figure 2). That is, the tool enables users to digest the original information in support of exploring data and generating hypotheses, and at the same time, interim discoveries are recorded using annotations, which could be represented in the annotation graph. Thus, the system should support exploratory analysis of an existing dataset, while also acting as an authoring tool to create a new artifact for further meta-analysis.

Under the discover goal, analysts *explore* the data since they usually do not know what or where the patterns are. Further, the system should provide various conceptual ways of turning the raw data into useful knowledge, including *identify*, *compare*, and *summarize* queries. To achieve these tasks, a user can perform the analysis of a dataset via *encoding*, *filtering*, *navigating*, *selecting*, *aggregating*, and *arranging* data observations. Under the produce goal, a user *annotates* the selected data observations to pose questions and create hypotheses, and further *records* them along with the dataset, which can be displayed with the annotation graph visualization to advance meta-level analysis.

### 3.3 Design Process

Our goal was to design and implement the proposed annotation graph—a dynamic visual exploration tool of the annotation semantics that include comments, tags, and data selections as well as their relations—within a single data annotation system.

Over the course of five months, we conducted multiple interviews with our experts, during which we discussed our design ideas, presented the latest working prototype, and collected their feedback on further improvements. We also exchanged emails regularly with the experts to keep them updated. As our main focus, the majority of the design process was on iteratively refining the annotation graph in C8. The major milestones of the process follow.

**Problem identification.** As the first step, we aimed to understand each research area that our experts focused on and extracted the common ground about why and how they explored their experiment data. We conducted a number of semi-structured interviews with them to identify the specific research questions that they wanted to answer, the characteristics of their datasets, the types of analyses that they usually perform, and the ways that they made sense of the data including generating hypotheses, documenting observations, and so on.

We obtained several high-level user requirements, which aligned with the theoretical principles outlined in our synthesis of ESDA and the multi-level task typology.

- *Supporting a fluid, integrated workflow of exploratory data analysis and making annotations.* This covers the key goals of ESDA and tasks in the typology. The system should smoothly combine both aspects, since meaningful annotations build upon effective data exploration.
- *Visualizing data in domain-specific context.* To allow for better sense-making in data exploration, the system should present observations in a domain-specific form that is familiar to the analyst.
- *Enabling rich, flexible annotation behaviors.* Analysts should be able to create annotations with any data objects (meaningful chunks), e.g., a time interval, an observation, and attributes across observations. The annotations could be light-weight tags or free-text notes.
- *Empowering meta-analysis through exploring relations in annotations.* Annotations should be used for more than just revisiting what has been found, but for promoting high-level analysis to indicate new directions and discoveries.

**Designing initial prototype.** Based on the previous results as well as the theoretical foundation, we developed the first prototype of C8 that supported a basic workflow of exploring and annotating our experts’ datasets. Inspired by our previous open-source tool [5], the prototype consisted of five coordinated views similar to the final prototype (Figure 3). A user can select a subset of data in the Data Grid, explore time-series cells in detail in the Timeline View, and further create and browse annotations in the Comments View. We implemented domain-specific Context Views (e.g., 3D curves, or heatmaps on a

trackpad) to visualize the experimental setup of the three areas. We presented the annotation graph concept using a node-link diagram with a traditional force-directed layout (nodes are pulled closer by edges).

We then conducted a series of interviews and observations with the experts to gather their feedback on this prototype. They were all satisfied with the flow of selecting data, exploring data, and making annotations (tags and comments). Moreover, they thought that it was useful to document their discoveries in such a data-driven and data-embedded way. However, they had difficulties understanding the force-directed layout, and thus could not use the annotation graph effectively. One expert mentioned: “[Nodes] constantly jumping around is not helpful and I’m not sure what the force conveys here.”

**Re-designing annotation graphs.** In this step, we re-designed the annotation graph. We replaced the force-directed layout with three different layout methods that enabled users to explore the annotation semantics from different perspectives, including a projection-plot, a slice-plot, and a circular-plot (see Section 4.2 for details). To reflect the annotation semantics, we developed metrics for measuring pairwise node similarities, which were used to inform the layout algorithms. In addition, we added features based on the experts’ suggestions in the previous iteration, such as searching annotations, and adding/removing tags to multiple comments at the same time.

We deployed this version for our experts to use during a week, and interviewed them to assess the strengths and weaknesses of the system. The experts liked it much better than the previous prototype, agreeing that the annotation graph was useful for browsing the existing annotations and potentially revealing new insights. However, they found that the graph layout sometimes did not align with their mental map of the annotations, for example, “I need these nodes to stay together, but [the graph layout method] doesn’t want to, and I have no way to change it.” One expert also requested more flexibility for color-coding time-series, such as encoding data points using another data attribute.

**Enhancing annotation graphs.** Building upon the most recent feedback, we developed a mixed-initiative approach for computing the annotation graph layout. That is, all nodes are positioned automatically based on their similarities, and a user could choose to manually override the position of specific nodes that are then integrated into the automatic layout. Moreover, we implemented a node merging feature that allows for a user to group nodes together as a single unit. We provided an updated version to the experts and invited them to further experiment with the new functionality. We then carried out short interviews with the experts, and made several small improvements to the prototype such as adding a configuration dialog to adjust the parameters of the automatic graph layout algorithm.

**Release and test.** In this phase, the final C8 prototype was deployed to the experts for two weeks, during which they could use the system freely to explore the data and make annotations. We instructed them to explore and annotate the data to identify interesting findings using the annotation graph. After, we conducted in-depth interviews to assess the effectiveness of the annotation graph within C8. Details of the results will be reported in Section 6.

## 4 ANNOTATION GRAPHS

In this section, we present the final annotation graph design that promotes the meta-analysis based on user-authored annotations. Other C8 interface components will be introduced in the next section.

### 4.1 Visual Design

The Annotation Graph View (Figure 3-e) helps an analyst visualize and explore the annotation semantics that they continuously generate during exploratory data analysis. Unlike the traditional approach of presenting user-authored annotations in a textual form, the annotation graph shows the dynamic relations between selections, comments, and tags (record, Connections). Thus, in the graph, color-coded nodes represent these three key elements in user annotations, and links are drawn if they appear in the same annotation. The node size is mapped to the number of connections, revealing the most frequently discussed concepts in all annotations.



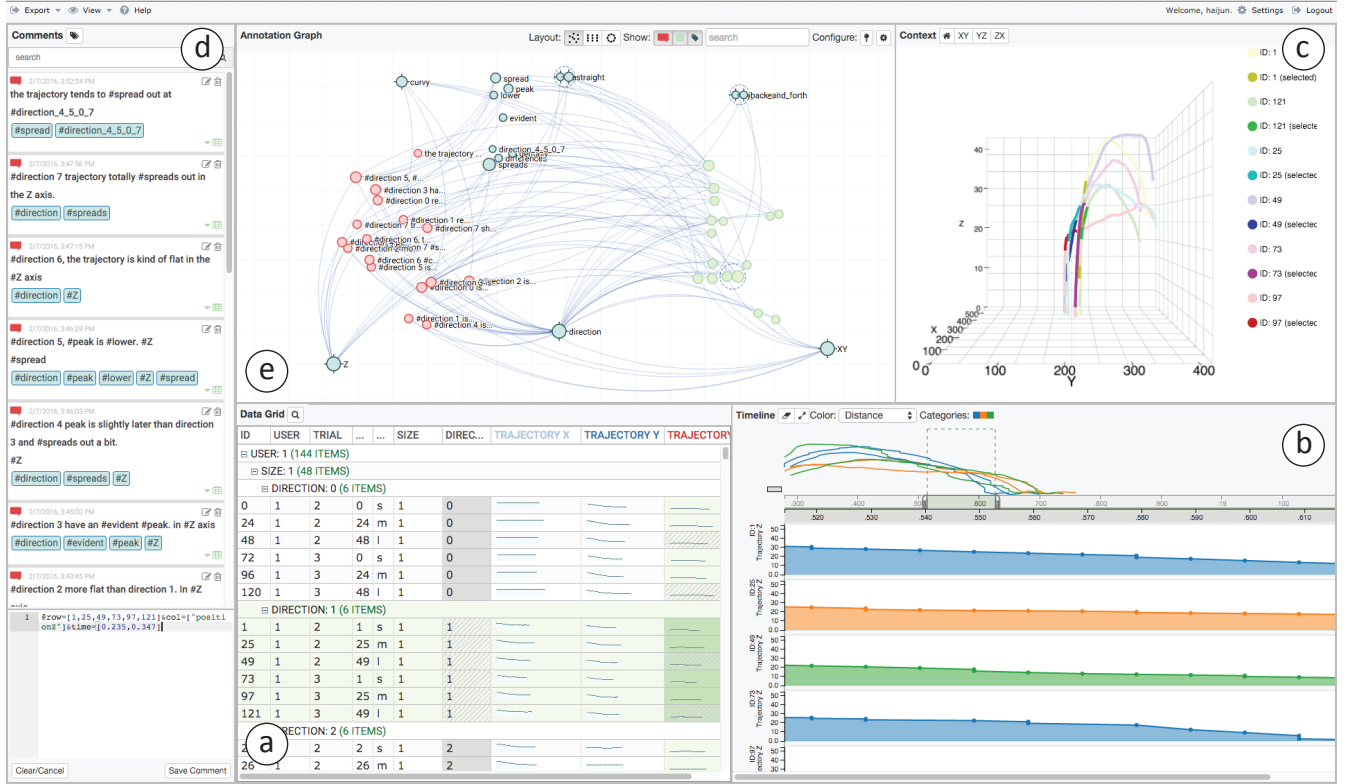


Fig. 3. The front-end interface of the C8 data annotation system: a) Grid View, b) Timeline View, c) Context View, d) Comments View, and e) Annotation Graph View. This particular snapshot features the use of C8 to analyze the results of an HCI user study that records participants pointing at a target on a tabletop display with different experimental conditions.



Fig. 4. Visual encodings used in the Annotation Graph View.

Three graph layout methods are provided to support viewing the annotation semantics from different perspectives (Figure 1), including a *projection-plot*, a *slice-plot*, and a *circular-plot*. In the projection-plot, nodes are positioned based on a global optimization of pairwise similarity between nodes, resulting in similar selections, comments, and tags clustering closer together. In the slice-plot, nodes are split by type with the horizontal positions restricted, but the vertical positions are determined by the similarity metric within each node type. In the circular-plot, a central node of interest is selected by the analyst, forcing similar nodes to reposition closer to the center. Each layout method has its own benefits, and complements the other layouts. For example, the projection-plot is useful to get a global picture; the slice-plot is clear for investigating connections among different types of nodes; and the circular-plot is helpful to reveal pairwise relations and similarity with respect to a particular node of interest.

We support user input to influence the organization of the nodes through a mixed-initiative approach for computing the graph layout, leveraging both the power of automatic similarity-based layouts and the mental model of an analyst. In the annotation graph, nodes can be switched to a manually-positioned mode, i.e., pinned on the canvas. These pinned nodes are constraints applied to the automatic layout methods, which affect the final location of all other nodes, resulting in a “magnet” metaphor pulling similar nodes towards them. As shown in Figure 4, we design a visual language to represent different characteristics of each node. Multiple nodes can be merged together using drag-and-drop actions, which is useful for grouping concepts in the annotations. We also support basic interactions such as filtering by node type, interactive searching, and tooltips on hover.

## 4.2 Similarity Measures and Layout Algorithms

To compute the layout in the annotation graph, we define a pairwise similarity measure for all types of nodes, including selections, comments, and tags. The measure contains within-concept and between-concept similarities. Let the annotation graph be  $G = (V, E)$ , where  $V$  is a set of nodes  $v_i$  and  $E$  is a set of edges  $(v_i, v_j)$ . We define the between-concept similarity (i.e., tag-comment, comment-selection, and selection-tag) based on their connections in graph:

$$S_{between}(v_i, v_j) = \begin{cases} w & (v_i, v_j) \in E \\ 0 & (v_i, v_j) \notin E \end{cases} \quad (1)$$

where  $w$  is a constant and  $v_i, v_j$  are two nodes (different types). We use 0.7 as the default value for  $w$ .

We further define three within-concept similarity metrics (i.e., selection-selection, comment-comment, and tag-tag) as follows.

- The selection-selection similarity is determined by how much two selections overlap in terms of rows, columns, and time intervals:

$$S_{selection}(s_i, s_j) = \alpha D_{row}(s_i, s_j) + \beta D_{col}(s_i, s_j) + \gamma D_{time}(s_i, s_j) \quad (2)$$

where  $\alpha$ ,  $\beta$ , and  $\gamma$  are weights,  $D(a, b) = |a \cap b| / |a \cup b|$  applied over the row, column, and time aspects of  $s_i, s_j \in V$  that are selection nodes. We set the three weights equal to 1/3 as the defaults.

- The comment-comment similarity is based on the bags-of-words model, in which each comment is represented as a vector of frequencies of its words [25], therefore:

$$S_{comment}(c_i, c_j) = \cos(\text{BoW}(c_i), \text{BoW}(c_j)) \quad (3)$$

where  $\text{BoW}$  is the bags-of-words transformation and  $c_i, c_j \in V$  are comment nodes.

- The tag-tag similarity is computed based on three factors, i.e., the word semantic meanings, characters, and co-occurrences in comments:

$$S_{tag}(t_i, t_j) = \alpha \cos(\text{Vec}(t_i), \text{Vec}(t_j)) + \beta \text{Ch}(t_i, t_j) + \gamma \text{Co}(t_i, t_j) \quad (4)$$

where  $\alpha$ ,  $\beta$ , and  $\gamma$  are weights (default values 1/3) and  $t_i, t_j \in V$  are tag nodes.  $\text{Vec}(t)$  is a transformation of the tag to its vector representation in the word semantic space based on the GloVe model [27]. We use the pre-trained mapping of 6B tokens based on Wikipedia and Gigaword.  $\text{Ch}(t_i, t_j)$  computes the similarity of two tags based on

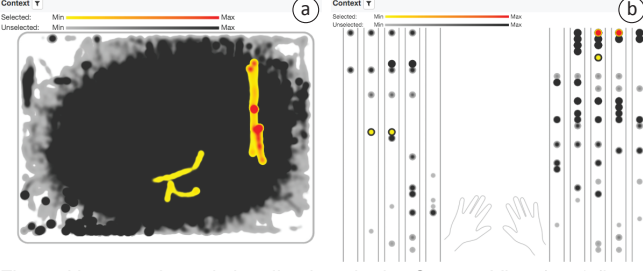


Fig. 5. Heatmap-based visualizations in the Context View for a) finger movements and b) finger chording patterns on a trackpad.

the Dice’s coefficient of their bi-gram character sets (e.g., “error” is represented as {er, rr, ro, or}) [25], i.e., the more common bi-grams they share the greater the similarity. The last factor  $Co(t_i, t_j)$  represents the normalized co-occurrences of the two tags in all the comments.

With the above measures, we derive a similarity matrix containing pairwise similarity of all the nodes in the graph, which is further used for layouts. Overall, the annotation graph layout follows a three-step process: 1) initially positioning all the unpinned nodes based on their similarity, 2) adjusting the positions based on their similarity to the pinned nodes, and 3) applying a repulsive force to reduce node overlaps.

The three layout methods impose different constraints on the node positions (Figure 1). In the projection-plot, the first step of the layout process employs the multidimensional scaling (MDS) algorithm [3] to place the unpinned nodes in 2D space, based on the similarity matrix. In the slice-plot, the MDS is only applied within each node type for positioning the nodes along the vertical axes. In the second step of the layout process, unpinned nodes are pulled by the pinned nodes based on their similarity and distance. The adjustment vector of an unpinned node  $v$  is defined as:

$$\Delta Pos(v) = \alpha \sum_i S(v_i, v) (Pos(v_i) - Pos(v)) \quad (5)$$

where  $v_i$  represents every pinned node,  $S(v_i, v)$  is one of the above similarity function,  $Pos$  is the location vector of a node, and  $\alpha$  represents the strength of this effect (default value 0.8). In the slice-plot, this effect is limited to the vertical positions of the unpinned nodes. The circular-plot places the most constraints on the nodes. In this representation, the node positions purely depend on the similarity between the central node and others. That is, the second and third steps of the above process do not take effect for positioning the nodes, and the pinned nodes are also moved by the automatic layout.

All the above parameters of the similarity and layout computation can be configured dynamically using a configuration dialog.

## 5 C8 INTERFACE

In this section, we describe the rest of the C8 interface (Figure 3). The Grid, Timeline, and Context Views support the Exploration Task, while the Comments View supports the Annotation Task (Figure 2). The back-end of the system (extended from our open-source tool [5]) provides a set of APIs that handle retrieving and storing the datasets, annotations, annotation graphs, and system configurations in MongoDB.

### 5.1 Grid View

The dataset under investigation is displayed in a tabular form in the Grid View (Figure 3-a). The Grid View aims to provide an overview of the *logical structure* of the data to facilitate data selections across rows and columns (navigate+select, Chunks). Each row represents a single observation record (e.g., a participant’s trial in an HCI experiment), with columns defining specific attributes of the record (e.g., experimental condition, data recorded in a trial). Cells in the grid can represent any type of data (including numerical, categorical, ordinal, and textual values), or entire time-series (including continuous values and discrete events) that are summarized using interactive sparklines [32]. A configuration schema describes the meta-data of each column and how it is displayed (encode, Conversions), and thus supports loading arbitrary time-series datasets.

Data displayed in the Grid View can be sorted and filtered based on the values of any column or combination of columns (filter, Constraints). Moreover, rows can be hierarchically grouped together according to column attributes. With the above manipulations to the data table, sparklines over columns support quick overview comparisons of the time-series data (arrange, Comparisons), for example, investigating the performance differences of participants under the same experimental condition to rule out outliers. Any combination of rows, columns, and individual cells can be selected (navigate+select, Chunks). The selected cells are highlighted in green and cells with existing comments are highlighted with a gray background.

### 5.2 Timeline and Context Views

The Timeline View (Figure 3-b) reveals the *temporal structure* of data and allows for data selections across time (navigate+select, Chunks). It shows details of time-series data selected in the Grid view, supporting further refinement of selections by specifying intervals of time-series (filter, Constraints). The Timeline View supports an overview and details approach of browsing the data. All the selected time-series are displayed using superposition in the global context chart on the top. An analyst can then specify a time window to browse individual time-series separately in juxtaposed detail views below (navigate+select, Chunks). In this way, a user can explore the time-series without visual clutter, synchronized with the rows in the Grid View. Further, the data points can be color-coded by any column attributes (discrete or continuous) in the Grid View (encode, Conversions), allowing an analyst to inspect the temporal correlations of multiple variables.

The Context View displays a contextual visualization for a specific dataset, which represents the *spatial structure* of the data and supports data selections from this perspective (navigate+select, Chunks). In our study, we developed three contextual visualizations for our experts, including a 3D view of user hand trajectories (Figure 3-c), a heatmap of finger touch points on a trackpad (Figure 5-a), and a heatmap of finger chording gestures with one pattern per row (Figure 5-b). Although these visualizations were designed to meet the specific needs of our expert users, any kind of custom visualization can be added to C8 using the configuration schema. The Context View updates when selections are changed in the Grid View (i.e., rows, columns, or cells) and the Timeline View (i.e., intervals). It further highlights the selected intervals of the data, with respect to the whole time-series. For example, the yellow-red color scale shows the selected interval, in contrast to the remaining data shown using a white-black color scale (Figure 5). This automatic aggregation of all the selected data elements provides a more concrete visual representation of the selection (aggregate, Computations).

Both the Timeline and Context Views support the detailed exploration of datasets to uncover features and patterns of interest. Together with selections in the Grid View, analysts can isolate regions of data important to their analyses, and quickly compare them at a granular level (arrange, Comparisons). For example, one can group the observations in the Grid View by the task repetition number in an experiment, select each repetition, and browse the aggregated visualizations in the Timeline and Context Views, to investigate potential learning effects in the study.

### 5.3 Comments View

Once interesting patterns are discovered in the data, an analyst can make annotations. The Comments View (Figure 3-d) manages the creation and modification of annotations. The annotations can be in the form of free-text comments and word tags (annotate, Comments+Codes) about data selections (navigate+select, Chunks). An analyst can author the annotation using the editing window at the bottom of the Comments View using a specific syntax, i.e., @selection and #tag (Figure 6). The syntax for the current selection is automatically inserted into the editing window, speeding up annotation. As an analyst gains experience and becomes familiar with the language, they can choose to manually edit or input the selections using the syntax. The corresponding visualization of this annotation in the Annotation Graph View is shown in Figure 4.

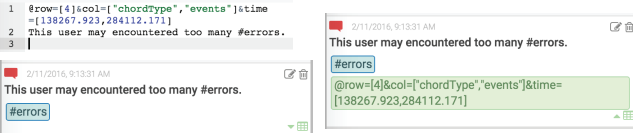


Fig. 6. Creating an annotation with a specific syntax, and the resulting annotation in the Comments View (with and without the selection hidden).

- Q1 What were your questions about the data? What did you try to identify?
- Q2 How did you perform the exploration and annotation of your data?
- Q3 What were your findings? How did the annotation graph help you (or not) in the process?
- Q4 Could you explain what you found with the annotation graph?
- Q5 Do you have any other feedback about the whole system?

Table 2. Themes of questions in the post-study interview.

Existing annotations are listed in reverse-chronological order, which can be revised when needed. By default, each card in the Comments View displays the comment and tags of the annotation; the selection syntax is hidden by default but can be revealed (Figure 6). When an annotation is selected, the Grid, Timeline, and Context Views automatically update to the visual state when the annotation was created, supporting a “bookmark” like feature to help an analyst go back to the original context of the annotation. A search bar further supports filtering the existing comments by keywords, tags, or selections, where query terms can be combined either in union or intersection logical modes, using commas and spaces, respectively.

Naturally, as an analyst explores and annotates their data, mental concepts emerge or change. For example, when encountering the same keyword again and again after creating multiple annotations, an analyst may want to promote that keyword into a tag. A “bulk tagging/untagging” function enables changes across multiple comments.

## 6 EVALUATION

In this section, we report the procedure and results of the “release and test” phase in our iterative user-centered design process (Section 3). The domain experts who participated in the design process were engaged again. This study aims to assess the utility of annotation graphs in helping users perform meta-analysis of data based on their annotations. We also collected general feedback on the design and usability of C8.

### 6.1 Method

The C8 system was deployed for a two-week field test with our three domain experts. Before the deployment, we presented the important system features as a refresher. We instructed each expert to “Use C8 to analyze your data and document findings that are valuable for your research” and “Use the annotation graph to organize your thoughts during the exploration and prepare to tell us a story about your findings.” To assess the potential of annotation graphs in a real-world scenario, we did not instruct them how to conduct their analysis, nor what to search for. During the deployment, we communicated with the experts via emails to resolve any issues about the interface. After the deployment, we conducted semi-structured interviews to learn about experts’ usage of the annotation graph, and collect general feedback on C8. The interview questions were along the themes in Table 2. For post-analysis, we recorded the interview sessions and exported their data annotations.

We chose this qualitative approach since our goal was to understand how annotation graphs are used during data exploration and meta-analysis. The combination of field test and interviews provided richer and deeper insights into the user experience.

### 6.2 Results

Overall, our experts agreed that the annotation graph played an important role in making sense of the data, relating evidence and questions, and organizing their thoughts, thus promoting higher-level insights. They thought that C8 was useful in generating hypotheses and documenting findings during exploratory analysis. In total, they created 107 annotations containing 63 unique tags and comments of 903 words, linked to 105 unique selections. We refer to our experts who analyzed the datasets described in Section 3.1 as E1, E2, and E3, respectively.

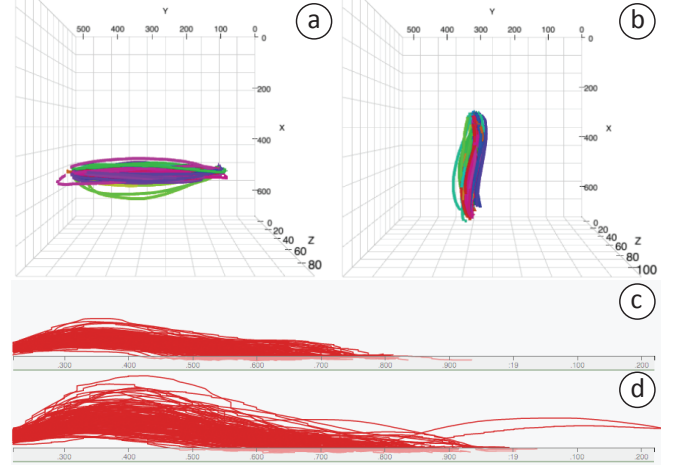


Fig. 7. The Context View shows that trajectories on the xy-plane are a) “more straight” or b) “curvier.” The Timeline View shows that trajectories on the z-axis are c) “less spread-out” or d) “more spread-out.”

#### 6.2.1 Analytical Processes

Although we observed different specific strategies of exploring and annotating the data, the general workflow was the same: conceiving initial questions with vague goals, examining the data and annotating parts that might be of interest, performing data meta-analysis from the annotations and generating explanations. This process could be iterative, in which the findings might elicit new questions and lead to further data explorations. During meta-analysis, we noted that each expert used different aspects of the annotation graph to manage their ideas and consolidate findings. Below we describe our experts’ specific usages of the annotation graphs and other functions in C8.

**3D Pointing Trajectories.** This expert (E1) had an initial idea that the 3D shapes of users’ hand trajectories may correlate with the target direction. He picked two interesting trajectory features to explore: curvatures on the xy-plane and variances along the z-axis, for each target direction. To analyze the data, he grouped the records by direction in the Grid View, and then selected every direction group to explore the trajectory time-series in the Timeline and Context Views. With the help of the visualizations that aggregate all selected trajectories, he was able to observe aspects (i.e., curvatures, variances) of their overall shape (Figure 7). Next, E1’s strategy was to annotate any features that might be interesting, resulting in a tag for the curvature (e.g., #curvy or #straight) and a tag for the variance (e.g., #spread). Most of the comments started with target direction (e.g., “direction NW...”) and E1 sometimes tagged the observed plane or axis (e.g., #XY, #Z). Also, he occasionally explored trajectories within a direction group, for example, by target size, when he noticed interesting patterns.

After going over all the directions, E1 employed different layouts of the annotation graph to perform meta-analysis, in which the slice-plot was used most often. An interesting finding was that E1 relied on the topology of the annotation graph rather than its spatial layout. This might be the reason why he liked the slice-plot since “It clearly splits different types of nodes and shows the connections.” He sometimes used the circular-plot for cross-referencing but found the projection-plot a bit overwhelming. In each layout, E1 filtered out selection nodes because “It generates too many links and the data itself is not important after it is tagged.” He also manually positioned certain tags to separate relevant concepts in his investigation. Through interacting with the annotation graph, E1 was able to identify several interesting findings in the dataset (Figure 8-a). For example, he observed that certain directions (N, NW, and S) had curvier trajectories whereas others (W, SW, E, and NE) were straighter. “This may reflect some patterns of human arm motor constraints, which is interesting to explore more,” he explained. E1 also found that trajectories on the E, SE, W, and SW directions had more variance, which seemed to have some correlation with the straightness. The above observations, revealed by the annotation semantics in graphs, encouraged him to do advanced analysis to confirm his findings.



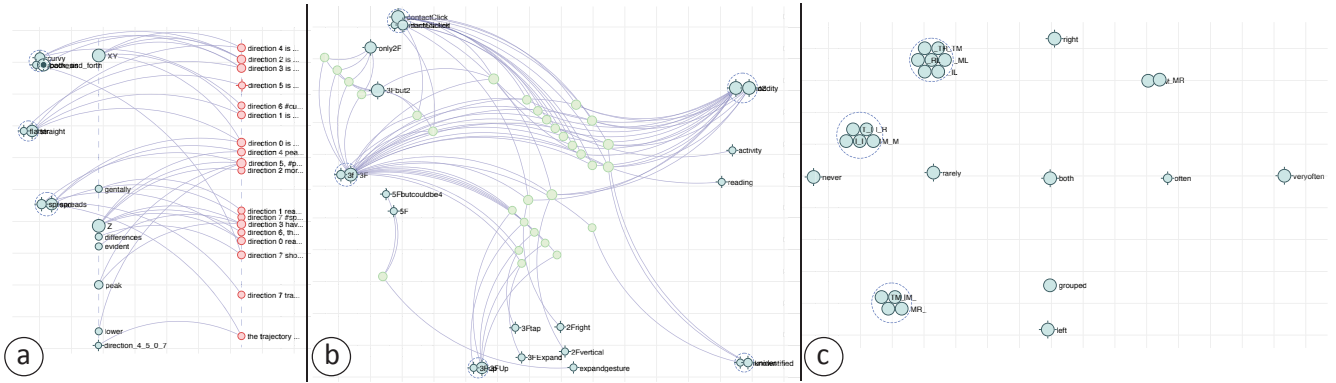


Fig. 8. Snapshots of the resulting annotation graphs created by our expert users during the deployment study: a) 3D hand trajectories, b) touchpad interactions, and c) finger chording gestures.

**Touchpad Interactions.** The general goal of the expert (E2) was to explore patterns of multi-finger gestures on the trackpad. He started by selecting all the events and used the Context View to identify that participants used the top-left and bottom-right corners of the trackpad infrequently (Figure 5-a). Using the sparklines of event series in the Grid View, E2 found that the majority of gestures were performed with one or two fingers, and there were far fewer gestures involving three-, four-, and five-finger gestures. He decided to explore three-, four- and five-finger gestures, because he expected that interesting patterns existed in gestures with more fingers. Hence, E2 selected the three-finger gesture row in the Grid View, zoomed-in the Timeline View to reveal the individual finger strokes, and created annotations by observing the Context View. He repeated the above procedure for every occurrence of the three-finger gestures. To facilitate the identification of individual finger strokes, he color-coded the timeline with the “event type” attribute (i.e., a single stroke would be a finger down, a series of finger moving, and a finger up, as shown in Figure 9-c).

During the exploration, E2 found that many of the recorded three-finger gestures were actually a two-finger scrolling with an additional static contact on the trackpad (Figure 9-a). He annotated these situations with tags like #3Fbut2 and #ContactClick to distinguish them from real three-finger gestures (e.g., #3F). Comment annotations often included gesture descriptions and potential activities of the participant, for example, “...#2FVertical movement (scrolling) with button hold. Must have been #reading at the same time.” After annotating the three-finger gestures, he explored the four- and five-finger gestures in a similar manner. To organize the annotation graph, E2 used the projection-plot to manually position tags into three groups—“physical contacts” on the left, “gestures” at the bottom, and “activities” on the right. After filtering out comment nodes, he used the layout of the annotation graph (Figure 8-b) to identify clusters of selection nodes. E2 mentioned that “I can clearly see the number of occurrences [of selection nodes] and the clusters and how they relate to different concepts.” One obvious finding was that “There were a lot of false three-finger gestures,” and he added “It is necessary to build a classifier to process the data before analyzing it.”

**Finger Chording Gestures.** The expert (E3) wanted to investigate common finger chords used by participants for particular gestures. After aggregating data by gesture type in the Grid View, he observed that nearly 95% of the interactions were scrolling. Using the Context View, E3 was able to get an overview of the chords people used most frequently during the experiment (Figure 5-b): “Although all participants were right-handed, people did use left-handed chords very often.” He first explored the scrolling gesture since it accounted for the majority of the data. E3’s annotation strategy was different from the previous two experts, in which he only created tags (no free-text comments) that “mapped” data patterns to his own language. To organize the Grid View, he grouped the rows by attributes: hand, then finger chord, and then user ID. By observing the Timeline View, he tagged the selected data with frequency (e.g., #often), hand (e.g., #left), and finger chords (e.g., #.TR meant right thumb and ring finger, and #MR\_ encoded left middle and ring fingers). Additionally, he said

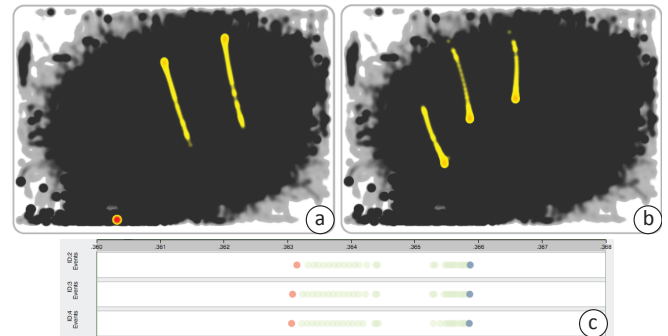


Fig. 9. a) A false three-finger gesture: two-finger scrolling with a static contact near the bottom-left. b) An actual three-finger gesture. c) Corresponding Timeline View color-coded with event type.

that “I never thought about the temporality [of finger chording gestures] before. It is an interesting research direction.” Thus, for some chords, he colored the event series with hand and was able to see if one hand was persistently used or two hands were interleaved.

To use the annotation graph for meta-analysis, E3 filtered out comments and selections and only looked at tag nodes, since all his annotations were tags. Using the projection-plot, he built a 2D scatterplot that embodied his mental model (Figure 8-c). He manually positioned frequency tags to create a horizontal axis (i.e., left to right: #never, #rarely, #often, and #veryoften) and hand tags to create a vertical axis (i.e., top to bottom: #right, #both, and #left). The remaining unpinned tags mostly represented finger chord codes. Leveraging the mixed-initiative layout, these finger chord tags automatically arranged into clusters along both axes. This ad-hoc scatterplot was created using the annotation graph and helped E3 investigate the nuances of finger chords that he was curious about. For example, he discovered a surprisingly large variety of chording gestures were performed to scroll; the most frequent ones were middle+ring and index+middle fingers of the right hand. Moreover, there was a greater variety of chords performed with the right hand, even for the same gestures, which might be because his participants were right-handed. E3 also explained he used the circular-plot to test the scatterplot that he built in the projection-plot: “It clearly shows the one-to-one relations [between tags]. I used it for debugging.”

## 6.2.2 General Feedback

**On Annotation Graphs.** Overall, all experts thought that the annotation graph helped them record, discover, and organize findings in their exploratory data analysis. “I did not use the Comments View much except making the annotations, because the graph already stores everything,” said E1. The other two experts also found the graph useful as “bookmarks” to revisit prior exploration contexts after clicking on nodes of interest. Since the Comments View updates accordingly when users interact with the annotation graph, the experts sometimes used the graph nodes as filters to show related annotations, thus facilitating the discourse when they presented their findings.

Additionally, all experts appreciated the node merging and pinning functions in the annotation graph. For example, E1 mentioned that “*The merging is very handy to unify some tags that actually mean the same thing, for example, #straight and #flatter,*” and E2 used the merging feature to correct typos or combine case-sensitive tags, commenting “*I don’t need to go back and change all the annotations.*” Overall, the experts felt that the pinning gave them control over the graph layout and facilitated the discovery by aligning to their mental models. Also, different purposes of using the pinning were observed, e.g., separating concept groups (E1, E2) and creating scatterplot axes (E3).

Feedback also addressed further improvement of the annotation graph. For example, E2 suggested the selection of multiple nodes and the support of hierarchical node merging. Completely filtering out certain nodes was also needed to simplify the visualization: “*Although the search bar can help on filtering, I need to disable some tags that are not useful, not letting them to affect the layout at all,*” said by E1.

**On C8 Interface.** The experts agreed that the design of C8 reflected their common workflows of analyzing data and could spark further ideation. E3 mentioned that C8 empowered his abilities of performing exploratory data analysis and authoring annotations. Since the annotation actions became quick and easy, he could tag as many data features as possible without thinking much, and then discover connections later with the help of the annotation graph. The experts also emphasized that at the end of the workflow they could use the visual and interactive features of the annotation graph as a communication aid, to walk others through the discoveries.

There were a few comments about enhancing the tool. For example, E2 thought that a multi-scale visualization in the Timeline View, rather than the overview and one-level detailed view, would make exploration easier. The experts appreciated the Context Views showing the general distribution of the selected data, but demanded richer interactions for more advanced visual analysis. They also mentioned that it took a long time to explore and annotate the data, and suggested that some automatic suggestions of tags or words for comments might help.

## 7 DISCUSSION

**Implications.** Throughout this design study, we learned a lot about creating effective annotation graph visualizations and advanced data annotation systems. First, the annotation graph needs to be flexible since analysts may be interested in different aspects of the graph to fulfill different purposes without being distracted by unnecessary data. For example, E1 used tag and comment nodes; E2 leveraged tag and selection nodes; and E3 only used tag nodes. E1 used only the slice-plot, whereas E3 used primarily the projection-plot and secondly the circular-plot. E1 most employed the topology of the graph to derive insights whereas E2 and E3 relied more on the layout. All experts thought that the flexibility of the annotation graphs was critical to exploring their annotation semantics, which they found to be one limiting factor of other data annotation tools.

Second, all the experts appreciated that they could control the graph layout using the node pinning function. Our mixed-initiative approach gave them the freedom and flexibility to choose a personalized balance (i.e., pin some) between fully automatic (i.e., pin nothing) and fully manual layout (i.e., pin everything). This was critical to reflect the mental models of the experts, and we observed different levels of controls of the graph layouts (Figure 8). E1 also suggested adding the ability to create a manual link between nodes to increase their similarities and bring them together, but he thought the node merging could achieve similar goals. Nevertheless, exploring manual & automatic linking support could be an interesting future direction.

Finally, while our experts enjoyed the meta-analysis process with the annotation graph, they generally thought that creating annotations was time-consuming, which is a critical step before meta-analysis. In some cases, it may be possible to lend support through a recommender system (e.g., E3’s annotation strategy, which coded attributes of the observations). In other cases, annotation activities conceivably require extensive human exploration of the data. This data exploration and annotation process could be valuable for further development of automated data coding scripts, rather than blind coding. For

example, E2 discovered the “false three-finger gestures” and planned to implement a classifier to distinguish them. Moreover, suggesting frequently-used or semantic-similar tags/terms for annotations might increase a user’s overall productivity. It is worth further investigating various computational methods to accelerate the annotation process.

**Limitations.** Although annotation graphs show promise to support meta-analysis in our study, several limitations were identified. First, the current graph visualization may not scale well for large numbers of annotations, as the nodes and links may become cluttered. In cases where topology is more important than layout, an adjacency matrix representation could be effective for dense annotation graphs. Second, node merging is currently limited to one level, and E2 suggested hierarchical merging could better reflect his mental model. Moreover, multi-level merging could potentially address the scalability issue. Multi-scale graph visualization techniques, such as PivotGraph [35], could be applied. Third, the mixed-initiative graph layout has several parameters that need to be tuned to best suit a user’s needs. Our experts thought that the layout parameters were too abstract, and they had difficulties in predicting the magnitude and direction of effect when changing parameters. Further research on a more intuitive interface for parameter adjustment is needed. Fourth, our deployment study used a qualitative approach to understand the ways users employed annotation graphs in their meta-analyses. A quantitative comparison of annotation graphs against existing annotation tools is left for future work.

**Generalization.** While annotation graphs and C8 were developed within the context of analyzing HCI experiment data, the overall framework is general and applicable to other datasets and domains. The design of the annotation graph is relatively independent of C8, and could be embedded into many other sensemaking and data annotation tools to support meta-analyses. Moreover, it is straightforward to include other sources of data in C8, for example, integrating eye-tracking observations similar to VA<sup>2</sup> [2]. Eye-tracking data is usually stored in the form of time-series or events which are already supported in C8, and the heatmap visualization in the Context View can be reused for showing fixations or gaze patterns. The development of custom contextual visualizations may be required for specific experimental setups. In addition to HCI experiments, we believe that C8 can be used in exploratory analysis and annotation of datasets from other application domains, because C8 supports a range of generic data types, including numerical, categorical, ordinal, and text attributes as well as continuous and discrete time-series.

## 8 CONCLUSION AND FUTURE WORK

In this paper, we have introduced the annotation graph driven by the mixed-initiative layout methods that allows analysts to externalize their thoughts, perform higher-level meta-analysis and support communication of findings in a dynamic and visual manner. We have also presented C8, an interactive visual data annotation tool grounded with the fundamental principles of ESDA and the multi-level typology of visualization tasks. We have evaluated C8 with three experts in a two-week deployment study. The results indicate that it is useful and effective in helping people perform exploratory analysis of the data, document their findings, and further discover and communicate thoughts with the annotation graph.

This work makes significant advances towards a better understanding of advanced visual data annotation systems and fleshes out several effective elements to better support data meta-analysis from user-authored annotation. Future research directions include the evaluation of annotation graphs with datasets from other application domains. We also plan to experiment with other visualizations for the annotation graph such as adjacency matrices, and to enhance its node merging capabilities. Moreover, we wish to enhance C8 by adding functions such as multi-level multi-scale exploration of time-series, and mechanisms for annotation recommendations.

## ACKNOWLEDGMENTS

This work was partially supported by Mitacs through the Mitacs Accelerate program and CPER Nord-Pas de Calais / FEDER DATA Advanced data science & technologies 2015-2020.

## REFERENCES

- [1] D. Billman, G. Convertino, J. Shrager, J. P. Massar, and P. Piroli. Collaborative intelligence analysis with cache and its effects on information gathering and cognitive bias. In *Human Computer Interaction Consortium Workshop*, 2006.
- [2] T. Blascheck, M. John, K. Kurzhals, S. Koch, and T. Ertl. VA<sup>2</sup>: A visual analytics approach for evaluating visual analytics applications. *IEEE Trans. on Visualization and Computer Graphics*, 22(1):61–70, 2016. doi: 10.1109/TVCG.2015.2467871
- [3] I. Borg and P. J. F. Groenen. *Modern Multidimensional Scaling: theory and applications*. Springer, 1997.
- [4] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):2376–2385, 2013. doi: 10.1109/TVCG.2013.124
- [5] S. Breslav. UberTagger: Exploratory analysis and annotation of sequential data [software]. Available from <https://github.com/sbreslav/UberTagger>, 2014.
- [6] S. Breslav, R. Goldstein, A. Khan, and K. Hornbæk. Exploratory sequential data analysis for multi-agent occupancy simulation results. In *Proc. Symposium on Simulation for Architecture & Urban Design*, pp. 59–66, 2015.
- [7] S. Breslav, A. Khan, and K. Hornbæk. Mimic: Visual analytics of online micro-interactions. In *Proc. the International Working Conference on Advanced Visual Interfaces*, pp. 245–252, 2014. doi: 10.1145/2598153.2598168
- [8] S. P. Callahan, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, and H. T. Vo. VisTrails: Visualization meets data management. In *Proc. ACM SIGMOD International Conference on Management of Data*, pp. 745–747, 2006. doi: 10.1145/1142473.1142574
- [9] Y. Chen, J. Alsakran, S. Barlowe, J. Yang, and Y. Zhao. Supporting effective common ground construction in asynchronous collaborative visual analytics. In *Proc. IEEE Conference on Visual Analytics Science and Technology*, pp. 101–110, 2011. doi: 10.1109/VAST.2011.6102447
- [10] G. Chin, Jr., O. A. Kuchar, and K. E. Wolf. Exploring the analytical processes of intelligence analysts. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pp. 11–20, 2009. doi: 10.1145/1518701.1518704
- [11] T. Chklovski, V. Ratnakar, and Y. Gil. User interfaces with semi-formal representations: a study of designing argumentation structures. In *Proc. ACM conference on the Intelligent User Interfaces Conference*, pp. 130–136, 2005.
- [12] E. C. Crowe and N. H. Narayanan. Comparing interfaces based on what users watch and do. In *Proc. the Symposium on Eye Tracking Research & Applications*, pp. 29–36, 2000. doi: 10.1145/355017.355022
- [13] W. Dou, D. H. Jeong, F. Stukes, W. Ribarsky, H. R. Lipford, and R. Chang. Recovering reasoning processes from user interactions. *IEEE Computer Graphics and Applications*, 29(3):52–61, 2009. doi: 10.1109/MCG.2009.49
- [14] M. Elias and A. Bezerianos. Annotating BI visualization dashboards: Needs & challenges. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pp. 1641–1650, 2012. doi: 10.1145/2207676.2208288
- [15] D. Gotz and M. X. Zhou. Characterizing users’ visual analytic activity for insight provenance. In *Proc. IEEE Symposium on Visual Analytics Science and Technology*, pp. 123–130, 2008. doi: 10.1109/VAST.2008.4677365
- [16] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Trans. on Visualization and Computer Graphics*, 14(6):1189–1196, 2008. doi: 10.1109/TVCG.2008.137
- [17] J. Heer and B. Shneiderman. Interactive dynamics for visual analysis. *Communications of ACM*, 55(4):45–54, 2012. doi: 10.1145/2133806.2133821
- [18] J. Heer, F. Viégas, and M. Wattenberg. Voyagers and voyeurs: Supporting asynchronous collaborative information visualization. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pp. 1029–1038, 2007. doi: 10.1145/1240624.1240781
- [19] T. J. Jankun-Kelly. Using visualization process graphs to improve visualization exploration. In J. Freire, D. Koop, and L. Moreau, eds., *Provenance and Annotation of Data and Processes*, pp. 78–91. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-89965-5\_10
- [20] Y. Kang, C. Gorg, and J. Stasko. Evaluating visual analytics systems for investigative analysis: Deriving design principles from a case study. In *Proc. IEEE Symposium on Visual Analytics Science and Technology*, pp. 139–146, 2009. doi: 10.1109/VAST.2009.5333878
- [21] Y.-a. Kang and J. Stasko. Characterizing the intelligence analysis process through a longitudinal field study: Implications for visual analytics. *Information Visualization*, 13(2):134–158, 2014. doi: 10.1177/1473871612468877
- [22] K. Luther, S. Counts, K. B. Stecher, A. Hoff, and P. Johns. Pathfinder: An online collaboration environment for citizen scientists. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pp. 239–248, 2009. doi: 10.1145/1518701.1518741
- [23] N. Mahyar, A. Sarvghad, and M. Tory. A closer look at note taking in the co-located collaborative visual analytics process. In *Proc. IEEE Symposium on Visual Analytics Science and Technology*, pp. 171–178, 2010. doi: 10.1109/VAST.2010.5652879
- [24] N. Mahyar and M. Tory. Supporting communication and coordination in collaborative sensemaking. *IEEE Trans. on Visualization and Computer Graphics*, 20(12):1633–1642, 2014. doi: 10.1109/TVCG.2014.2346573
- [25] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [26] T. P. Moran, W. van Melle, and P. Chiu. Spatial interpretation of domain objects integrated into a freeform electronic whiteboard. In *Proc. ACM Symposium on User Interface Software and Technology*, pp. 175–184, 1998. doi: 10.1145/288392.288600
- [27] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global vectors for word representation. In *Proc. Empirical Methods in Natural Language Processing*, pp. 1532–1543, 2014.
- [28] P. Piroli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proc. International Conference on Intelligence Analysis*, vol. 5, pp. 2–4, 2005.
- [29] P. M. Sanderson and C. Fisher. Exploratory sequential data analysis: Foundations. *Human-Computer Interaction*, 9(4):251–317, 1994. doi: 10.1207/s15327051hci0903&4\_2
- [30] Y. B. Shrinivasan, D. Gotz, and J. Lu. Connecting the dots in visual analysis. In *Proc. IEEE Symposium on Visual Analytics Science and Technology*, pp. 123–130, 2009. doi: 10.1109/VAST.2009.5333023
- [31] Y. B. Shrinivasan and J. J. van Wijk. Supporting the analytical reasoning process in information visualization. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pp. 1237–1246, 2008. doi: 10.1145/1357054.1357247
- [32] E. Tufte. *Beautiful Evidence*. Graphics Press, 2006.
- [33] J. W. Tukey. *Exploratory Data Analysis*. Pearson, 1977.
- [34] F. Viégas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon. ManyEyes: a site for visualization at internet scale. *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1121–1128, 2007. doi: 10.1109/TVCG.2007.70577
- [35] M. Wattenberg. Visual exploration of multivariate graphs. In *Proc. SIGCHI conference on Human Factors in computing systems*, pp. 811–819, 2006. doi: 10.1145/1124772.1124891
- [36] W. Willett, J. Heer, J. Hellerstein, and M. Agrawala. CommentSpace: Structured support for collaborative visual analysis. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pp. 3131–3140, 2011. doi: 10.1145/1978942.1979407
- [37] W. Wright, D. Schroh, P. Proulx, A. Skaburskis, and B. Cort. The sandbox for analysis: Concepts and methods. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pp. 801–810, 2006. doi: 10.1145/1124772.1124890
- [38] K. Xu, S. Attfield, T. J. Jankun-Kelly, A. Wheat, P. H. Nguyen, and N. Selvaraj. Analytic provenance for sensemaking: A research agenda. *IEEE Computer Graphics and Applications*, 35(3):56–64, 2015. doi: 10.1109/MCG.2015.50
- [39] J. Zhao, D. Wigdor, and R. Balakrishnan. TrailMap: Facilitating information seeking in a multi-scale digital map via implicit bookmarking. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pp. 3009–3018, 2013. doi: 10.1145/2470654.2481417